PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
(Case No. 99,447)

In the Application of:	)	
	)	
Michael S. Borella et al.	)	
	)	Examiner: Vaughan Jr., William C.
Serial No.: 09/511,735	)	
	)	Group Art Unit: 2143
Filing Date: February 24, 2000	)	
	)	Confirmation No.: 5494
For: Method and Application Programming	)	
Interface for Assigning Multiple	)	Customer No.: 20306
Network Addresses	)	

DECLARATON PURSUANT T 37 C.F.R. § 1.131

Dear Sir:

We, Michael S. Borella, residing at 1208 Haverhill Circle , Naperville, Illinois 60563, and Nurettin Beser, residing at 1781 Karameos Dr. Sunnyvale, CA, 94087, hereby declare:

1. We are the original and joint inventors of the subject matter in the United States Patent Application Serial No. 09/511,735, filed on February 24, 2000, and entitled, "Method and Application Programming Interface for Assigning Multiple Network Addresses."

2. The inventions disclosed in the above-captioned patent application were conceived cooperatively by Michael S. Borella and Nurettin Beser at least as early as October 5, 1999.

3. Accompanying this Declaration is Exhibit 1, which contains photocopies of an Invention Disclosure Form that was prepared and dated prior to October 5, 1999 and illustrates an initial conception of our inventions.

McDONNELL BOEHNEN
HULBERT & BERGHOFF LLP
300 SOUTH WACKER DRIVE
CHICAGO, IL 60606
TELEPHONE (312)913-0001

initial conception of our inventions.

4. The dates from Exhibit 1 have been redacted. However, the dates of Exhibit 1 are at least on or before October 5, 1999.

5. The inventions disclosed in the above-captioned patent application were conceived at least as early as October 5, 1999 and constructively reduced to practice at least as early as February 24, 2000.

6. Our inventions were conceived and diligently reduced to practice in the United States.

7. We hereby declare further that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: 11/14/05 Signed: _____
Michael S. Borella

Date: _____ Signed: _____
Nurettin Beser

McDONNELL BOEHNEN
HULBERT & BERGHOFF LLP
300 SOUTH WACKER DRIVE
CHICAGO, IL 60606
TELEPHONE (312)913-0001

4. The dates from Exhibit 1 have been redacted. However, the dates of Exhibit 1 are at least on or before October 5, 1999.

5. The inventions disclosed in the above-captioned patent application were conceived at least as early as October 5, 1999 and constructively reduced to practice on February 24, 2000.

6. Our inventions were conceived and diligently reduced to practice in the United States.

7. We hereby declare further that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: _____ Signed: _____
Michael S. Borella

Date: 11.14.2005 Signed: _____
Nurettin Beser

McDONNELL BOEHNEN
HULBERT & BERGHOFF LLP
300 SOUTH WACKER DRIVE
CHICAGO, IL 60606
TELEPHONE (312)913-0001

2459.

CS
U.S.P

99457

MEMO

This is an Information Disclosure Form for 3Com Corporation and all of its related companies, hereafter referred to as "COMPANY."

Date Stamp:

Instructions:

- Complete all sections. Attach supplemental pages and other materials.
- Provide enough information to understand your invention.
- Any questions regarding this form or how to fill it out, refer to Kim Clinger (408 326-1754) or Bill Becker (408 764-5485) by phone or e-mail.
- **Return Completed Form To:** Legal Department - M.S. 1308
  5400 Bayfront Plaza
  Santa Clara, California 95052-8145

1. Title of the Invention Write a short descriptive title, avoiding coined terms and project names.

Per-Application Virtual Interface Support for TCP/IP

2. Inventors Include names of all persons who made a contribution to the conception and/or reduction to practice of the invention. If there are more than five inventors, use a Supplemental Sheet.

Full Name:	**Michael S. Borella**	M.S. & Location	**UNET Cape Code North 45 N.114**
Home Address:	**1208 Haverhill Circle, Naperville IL 60563**		
Home Phone:	**(630) 961-3750**	Work Phone:	**(847) 342-6093**
Employee No.	**21032**	Division:	**Network Systems**
Supervisor:	**Ikhlaq Sidhu**	Department:	**Advanced Technologies**
Citizenship:	**USA**	:	
Full Name:	**Nurettin B. Beser**	M.S. & Location	**RM.105**
Home Address:	**628-2 Sheridan Square**		
Home Phone:	**(847) 864-6930**	Work Phone:	**(847) 262-2195**
Employee No.	**20016**	Division:	**PCD**
Supervisor:	**Phillip Robinson**	Department:	**Cable Access**
Citizenship:	**Turkish**	:	
Full Name:		M.S. & Location	
Home Address:			
Home Phone:		Work Phone:	
Employee No.		Division:	
Supervisor:		Department:	
Citizenship:		:	
Full Name:		M.S. & Location	
Home Address:			
Home Phone:		Work Phone:	
Employee No.		Division:	
Supervisor:		Department:	
Citizenship:		:	
Full Name:		M.S. & Location	
Home Address:			
Home Phone:		Work Phone:	
Employee No.		Division:	
Supervisor:		Department:	
Citizenship:		:	

Check here and attach supplemental sheet if more than 5 inventors ☐

3. Conception of the Invention

Date of conception: _____ Location of conception **3Com Rolling Meado**

Date of first written description: _____

Location of such first description **3Com Mt. Prospect** Page(s): _____

Please attach copies of all pertinent lab notebooks or equivalent. They should be signed, dated and witnessed by two other people who have read and understood the description of the invention.

4. Reduction to Practice

Reduction to practice is demonstrating that an invention merely works with evidence such as working models, prototypes or simulations. Reduction to practice is not necessary to file a patent application.

Date of any such demonstration: none planned ☒

date/planned date of demonstration _____

Location of demonstration: _____

5. Invention Applicability, Project, Release, Sale Information

A sale, offer for sale, public showing or release of the invention may affect COMPANY's right to patent the invention. Submit this form even if a public sale, showing or release has occurred.

To which division or operation does this invention best apply?	**Client software**
Field of technology (e.g., manufacturing, switches, hubs, routers, network management software, adapters, etc.)	**Adaptors, Interfaces**
Project name and description:	
Product name and model number:	
Does this invention relate to an actual or proposed standard or defacto standard?	☒ **NO** ☐ **YES -(name of std.)**
Please list docket numbers of all other invention disclosures that are related to this one:	
Estimated/actual date of first public release or showing of invention or product incorporating or using the invention:	
Estimated/actual manufacturing release date of invention or product incorporating or using the invention:	
Estimated/actual date of offer for sale of product incorporating or using the invention:	

6. Publication of the Invention

Publication of a description of the invention may affect COMPANY's right to patent the invention. Submit this form even if publication has occurred.

Has a description been published or is it scheduled to be published? No ☒ Yes ☐

If "Yes," when and to whom? _____

Has a description been disclosed or is it scheduled to be disclosed? No ☒ Yes ☐

If "Yes," when and to whom? _____

Was a Non-Disclosure Agreement used? No ☐ Yes ☐

If "Yes," please attach copy. _____

7. Government Agency Contract

Was this invention made under a government agency contract? No ☒ Yes ☐

If "Yes," government agency contract number: _____

8. Joint Development Contract

Was this invention jointly developed with inventors from another company? No ☒ Yes ☐

If "Yes," please identify the company and/or non COMPANY inventors: _____

Was the invention tested, constructed or conceived pursuant to the performance of a development contract with another company? No ☐ Yes ☐

If "Yes," please identify the contract and its location: _____

9. Related Art

Is this invention an improvement of an existing COMPANY product? No ☐ Yes ☐

If "Yes," identify the existing product: _____

Closest known related art ("prior art"): _____

What was the problem to be solved? _____

How had others attempted to solved it before you? _____

What were the problems or disadvantages with prior solutions? _____

10. Keywords

Indicate any key words (preferably at least 3) we could use to search for related art or identify this invention for use in our own database. **Application programming interfaces, virtual TCP/IP interfaces**

11. Background Materials

List any printed publications, patents, patent applications or any other materials you are aware of which provides background material and/or prior art for your invention _____

12. Brief Description of the Invention

Describe the structure, function and/or method of the invention in just enough detail to enable someone technical to understand your invention. Stress the fundamental principle of the new idea from an engineering standpoint. Attach all the relevant descriptive materials. You should reference/include any drawings/sketches that will help explain the invention.

13. Drawings of the Invention

Please submit clear drawings and/or sketches which illustrate the invention either by electronically inserting them into Section 12 or using the supplemental sheets if you can't easily electronically insert them.

14. Licensing Considerations

Is this a licensable technology? No ☐ Yes ☒ Don't know ☐

If "Yes," name the fields in which this might be licensed: _____

If "Yes," name the companies which may possibly be interested: _____

15. Attorney

If there is a particular patent attorney with whom you would like to work on this disclosure, suggest his/her name. _____

16. Signature(s)

This Invention Disclosure Form is submitted pursuant to your employment agreement with COMPANY. Use a Supplemental Sheet if there are more than 5 inventors. Please sign and date below and be certain that each page of this disclosure has been initialed by each inventor.

Signature: _____ **Date:** _____

Signature: _____ **Date:** _____

Signature: _____ **Date:** _____

Signature: _____ **Date:** _____

Signature: _____ **Date:** _____

17. Witnesses: Read and Understood

This Invention Disclosure Form consisting of pages has been read and understood by:

Name: _____ **Name:** _____

Signature: _____ **Signature:** _____

Date: _____ **Date:** _____

Witnesses, please initial all supplemental pages.

2459

This is a SUPPLEMENTAL INVENTOR SHEET which is to be used if there are more than 5 inventors for the invention set out in the Invention Docket specified above.

2. Additional Inventors: Include names of all persons who made a contribution to the conception and/or reduction to practice of the invention. If there are more than five inventors, use a Supplemental Sheet.

Full Name: M.S. & Location
Home Address:
Home Phone: Work Phone:
Employee No. Division:
Supervisor: Department:
Citizenship:

Full Name: M.S. & Location
Home Address:
Home Phone: Work Phone:
Employee No. Division:
Supervisor: Department:
Citizenship:

Full Name: M.S. & Location
Home Address:
Home Phone: Work Phone:
Employee No. Division:
Supervisor: Department:
Citizenship:

Full Name: M.S. & Location
Home Address:
Home Phone: Work Phone:
Employee No. Division:
Supervisor: Department:
Citizenship:

Full Name: M.S. & Location
Home Address:
Home Phone: Work Phone:
Employee No. Division:
Supervisor: Department:
Citizenship:

2457

This is a SUPPLEMENTAL INFORMATION SHEET to be used to provide additional information regarding the invention disclosure referenced above.

Item ___

Per-Application Virtual Interfaces for TCP/IP Network Hosts

Abstract

Currently, a TCP/IP stack differentiates applications on the transport layer, typically with port numbers. In some scenarios, it is desirable to be able to differentiate applications based on a network-layer identifier; i.e., an IP address. In this document, we describe for per-application virtual interfaces can be added to a TCP/IP stack, and how the socket API can be modified for their support.

Revised :

Authors :

3Com
1800 West

Central Rd

3Com Invention Disclosure Form

3Com Confidential

IDF Docket No. _____

Introduction

Currently host TCP/IP stacks allow a single host to support multiple physical and logical data-link interfaces. A data-link interface, for purposes of this document, is a device that permits a host to communicate with some entity. For example, an Ethernet card, an ISDN card, a loopback interface, and a modem PPP connection are examples of data-link interfaces. These interfaces may be either hardware or a combination of hardware and software. They may be active for as long as the host is running, or they may activate and de-activate dynamically. For example, an Ethernet interface consists of the Ethernet card along with a device driver. Typically an Ethernet interface actives upon system boot and remains active until the host is shut off. A modem PPP interface, on the other hand, is present only when the modem is connected to a remote access server. A loopback interface does not have a hardware component – instead it consists entirely of software and allows a host to transmit to itself using IP address 127.0.0.1. Examples of the configuration of these interfaces for a host using an Ethernet and a loopback are shown in Figure 1.

```
eth0    Link encap:Ethernet  HWaddr 00:60:97:5F:1E:DA
        inet addr:149.112.60.14  Bcast:149.112.60.255  Mask:255.255.255.0

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
```

Figure 1: Example data-link interfaces.

Data-link interfaces represent an IP address that is bound to a data-link device. For example, in Figure 1, the Ethernet data-link interface eth0 is bound to IP address 149.112.60.14. All communication via these interfaces occurs through device-independent calls to the socket application programming interface (API). The socket layer exists logical above the transport layer, but below the application layer. This API is well defined in the UNIX on-line manual pages as well as many textbooks.

In new and emerging communication systems, it will be advantageous to allow per-application virtual data-link interfaces (PAVDLIs). A PAVDLI is different from a traditional data-link interface in that it represents an IP address that is bound to a given executing instance of an application (e.g., a process or related group of processes). Thus, more than one PAVDLI may map to the same physical or logical data-link device, but will only send and receive data on behalf of a single application.

PAVDLI's will be useful for differentiating IP data traffic to or from a particular host based on something other than a transport-layer parameter such as a TCP or UDP port. The particular application that we consider is a host dedicated to Internet telephony or multimedia services that resides behind a dual NAT (see below). Due to the fact that multiple sessions on the same host must be assigned different IP addresses, we require a method and API for binding multiple IP addresses to the same physical data-link interface, and being able to tell them apart from one another at the application layer.

Architecture

A dual NAT architecture consists of one or more private IP address spaces connected by a public IP address space. In an IP telephony environment, it is desirable that user agents (hosts) reside only in the private address spaces (i.e., network 10), so that IP addresses are not unnecessarily depleted. These hosts may take the form of broadband telephony interfaces (BTIs), which convert analog telephony signals from some number of RJ-11 jacks to packetized voice, and vice-versa.

Our goal in specifying this architecture is to provide an environment such that public IP addresses are never present on the private networks. When a private host transmits to the public network, it uses a local private proxy IP address. This address is owned by the private network's edge router (ER), which is also a NAT. The NAT routers perform double "traditional" NAT, in which no port translation is required, but both source and destination IP addresses are re-mapped for both outgoing and incoming packets. When two hosts on different private networks communicate, NAT is performed twice – once when the packets leave the first private network, and again when the packets enter the second private network. Call setup, teardown and billing signaling is facilitate through the use of gatekeepers (GK's) and back end servers (BES). An example architecture is show in Figure 3.

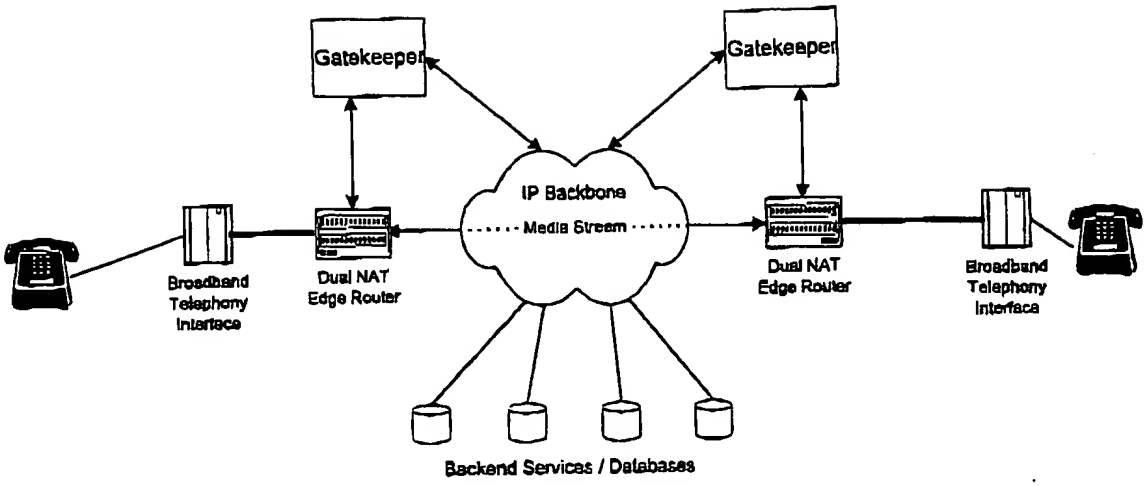

Figure 3: Example architecture for dual NAT public/private network.

Previous Work

The solution presented here uses a number of concepts described in Internet Draft <draft-ietf-nat-terminology-02.txt>, in particular, Traditional NAT, Two-Way NAT and Twice-NAT.

Terminology

We use the following terminology in our diagrams and discussion.

- BTI: Broadband Telephony Interface. Customer premises equipment that performs A/D and D/A conversion of voice signals. Provides one or more addresses in the private network space.
- ER: Edge Router. A layer-four router and NAT, which is located in on provider premises (e.g., a central office). Contains multiple private and public addresses and serves as a transparent gateway between the two address spaces.
- GK. Gatekeeper. Each gatekeeper is in charge of some number of edge routers. Gatekeepers communicate with one another using a signaling protocol, such as SIP or H.323, in order to set up and tear down calls. An edge router cannot open a media stream to another edge router without being instructed to do so by a gatekeeper.
- BES: Back end servers. An array of servers, possibly replicated throughout the VoIP network, that form a distributed database. The information that they contain includes customer records, billing data, and location services. A BTI must register itself with the BES before can be located by the network.

Stack Modifications

In order to support PAVDLIs, host TCP/IP stacks must be modified to be able to support multiple IP addresses per interface, as well as the concept of ephemeral addresses. Furthermore, the stacks must be able to be assigned these addresses dynamically, through a mechanism such as DHCP or IPCP. Alternatively, static assignment of addresses may occur. When a user application requests a new address to use, the stack must be able to request the address from a server, receive an address assignment, and associate that address with the socket.

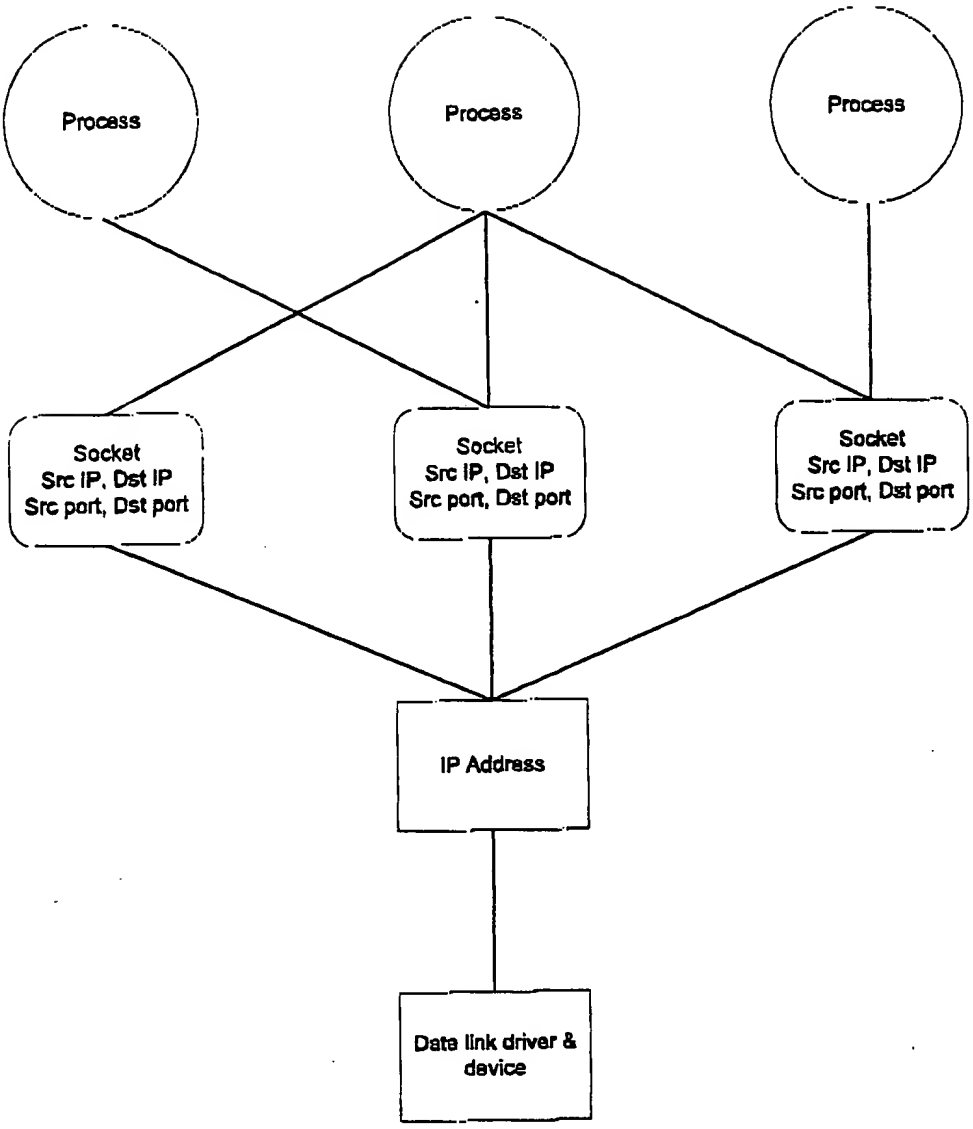

Figure 5: Current TCP/IP stack implementation.

Figure 5 shows the TCP/IP stack implementation that is used in many operating systems. One or more processes may each open one or more sockets. Each socket consists of source and destination IP addresses, as well as source and destination ports. Each of the IP addresses in the sockets are actually pointers to the host's IP address, which in turn is bound to a data-link interface. These IP addresses may be static or can be chosen by the kernel based on a routing decision – i.e., based on which interface the data will egress from. Figure 6 shows the TCP/IP implementation that would be used with PAVDLIs. Each socket may use a different IP address, and all IP addresses may be bound to the same interface.

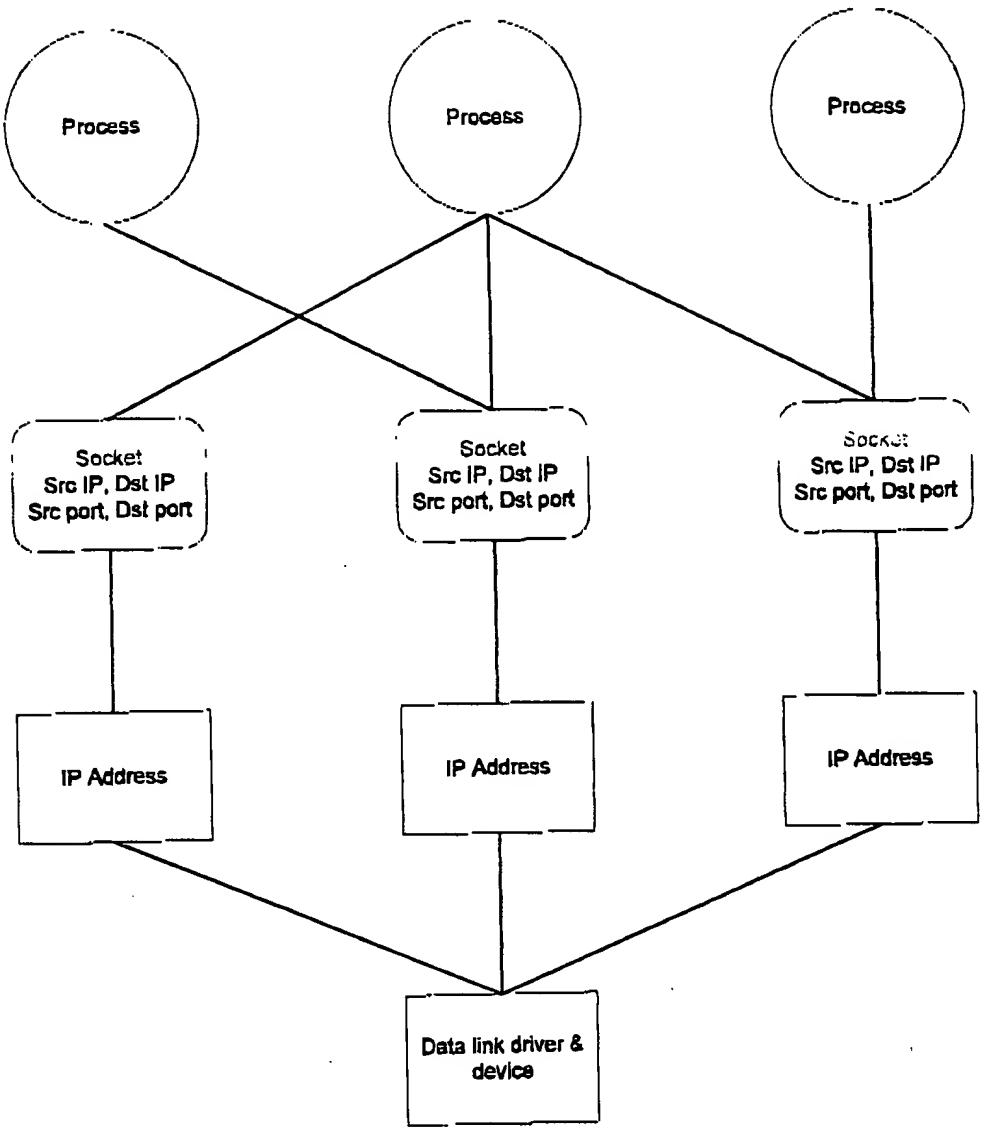

Figure 6: TCP/IP implementation for PAVDLL

API Modifications

The socket API must be modified so that users of PAVDLIs can do the following:

- Request that a new IP address be allocated for a particular socket. This can be done via modifications to the socket() system call.
- Bind the new IP address to the socket. This would require modifications to the bind() and connect() system calls.
- Based on the socket identifier, determine the value of the IP address associated with the socket. This will require modifications to the getsockname() function.

All other socket functions will remain the same.

Modifications to socket()

The socket() system call is currently defined to take three parameters:

- Address family: AF_INET, AF_INET6.
- Type: SOCK_STREAM, SOCK_DGRAM, SOCK_RAW.
- Protocol: Typically 0 except for raw sockets (i.e., sockets with a type of SOCK_RAW).

The call must be modified such that the stack is informed that a new IP address is to be used with this socket. A new parameter could be added to the call, or a new address family, such as AF_INET_PAVDLI could be define that acts

identically to AF_INET, except that when an AF_INET_PAVDLI socket is opened, the stack requests a new IP address from an address server.

Modifications to bind()

The bind() system call is typically used by servers and UDP clients. It is currently defined to take three parameters:

- Socket identifier: The socket identifier returned from a socket() call.
- Socket address structure: An IP address and a port number.
- Socket address structure length: The size in bytes of the socket address structure.

Servers typically supply a well-known port number to the socket address structure, but leave the IP address portion empty, so that the stack can fill it in. This is reasonable when a host only uses a single IP address, or when just one of the host's IP addresses needs to be published with the server. UDP clients must also bind an address and port to the socket, but typically do so by leaving both the IP address field and the port field blank, so that they can be filled in by the stack. The PAVDLI bind() call must always use the IP address associated with the socket identifier.

Modifications to connect()

The connect() system call is typically used by both TCP and UDP clients to have the stack allocate an IP address and an ephemeral port. It is currently defined to take three parameters:

- Socket identifier: The socket identifier returned from a socket() call.
- Socket address structure: An IP address and a port number of the server to contact.
- Socket address structure length: The size in bytes of the socket address structure.

A caller of connect() leaves it completely up to the stack to determine the local IP address and port number. Thus, the connect() call must be modified, like the bind() call, to always use the IP address associated with the socket identifier.

Modifications to getsockname()

The getsockname() system call is used by an application to determine the local IP address and port used in a socket. It is currently defined to take three parameters:

- Socket identifier: The socket identifier returned from a socket() call.
- Socket address structure: An IP address and a port number.
- Socket address structure length: The size in bytes of the socket address structure.

After getsockname() is called, the stack fills in the socket address structure with the IP address and port that is being used. The getsockname() call must be modified so that it always returns the IP address that has been associated with the socket.